

## IMPLEMENTING A FRAME REPRESENTATION IN CLIPS/COOL

Leonard Myers and James Snyder

CAD Research Unit, Design Institute  
California Polytechnic State University, San Luis Obispo

**Abstract.** The purpose of this paper is to describe and evaluate an implementation of frames in COOL. The test case is a frame-based semantic network previously implemented in CLIPS Version 4.3 as part of the Intelligent Computer-Aided Design System (ICADS) and reported in the first CLIPS conference.

### THE ICADS SYSTEM

The CAD Research Unit of the Design Institute at Cal Poly is engaged in a long term development of an Intelligent Computer Aided Design System (ICADS), (Pohl et al. 1989). Central to this work is the philosophy that the system should be a valuable assistant to the designer throughout the entire design activity. The current working ICADS prototype is a distributive system that can run with a variety of hardware processors in a Unix environment. It includes an Expert Design Advisor that has six knowledge based systems working as domain experts, a Blackboard Coordination Expert, two knowledge bases and several sources of reference data. The Expert Design Advisor in the prototype interprets a drawing as it is being made by a designer working in a CAD environment. It reacts in real-time to monitor the evolving floorplan from the viewpoints of experts in the domains of Access, Climate, Cost, Lighting, Sound, and Structure. The system also provides for additional interaction with the designer outside the CAD system to help realize a resultant design that is free from conflicts in the six areas listed.

### The Frame Representation

All architectural information that is used for inference in the ICADS model, including the project specification information, is put into a specific frame representation (Assal and Myers 1990). The inferences are made by expert systems written in CLIPS (NASA 1990), an expert system shell, so the frame representation is designed particularly for the form of CLIPS facts. The frame implementation consists of representation, generation, and manipulation features beyond that which will be described here. However, the most important ideas can be easily seen.

Each frame will either hold a class or an instance of a class. If the frame holds a class, it will describe the basic characteristics of the class including: default values; demons that dynamically obtain values or perform tasks for instances of the class; the names of the slots in the class; and relations between the class and other classes.

Table 1 shows a simple declaration of an instance of a 'wall' class. The frame is merely a collection of CLIPS facts that satisfy certain rules of form. The number '21' in the

(	FRAME	wall	21	)	
(	VALUE	wall	length	21	10
(	VALUE	wall	orientation	21	180
(	RELATION	wall	window	21	33
(	VALUE	window	width	21	36
(	VALUE	window	leads-to	21	16
(	VALUE	space	name	16	EXTERNAL)

**Table 1.** A Partial Instance of a 'wall' Class

example is an identifier for the particular instance of this wall class. The first slot shown holds the length of the particular wall. Another slot in the wall frame is denoted as 'orientation' and identifies the placement of the wall. Further, there is a relation to the 'window' class, which is interpreted to mean that a wall can have a window. As implied by these examples, the 'RELATION' fact is normally used to identify a 'has-a' relation. For example, '33' might be the instance identifier of a particular window that is in the wall. The wall also has the slots, 'width' and 'leads-to'. The final 'VALUE' fact shows that the window 'leads-to' the outside world, which is identified as 'EXTERNAL', through the name of the space instance referenced by '16', connecting the last two facts.

In addition to the above 'FRAME', 'VALUE', and 'EXTERNAL' keywords, there are also 'DEFAULT' and 'DEAMON' facts. The 'DEFAULT' fact can be used to establish values that are not set when an instance of a class is generated. For example, in the current ICADS system the ceiling-height slot of any instance of a space is 8 feet, by default. Finally, there is a 'DEAMON' that indicates actions that should be taken if certain conditions apply for the particular slot that is referenced. One of the options for the 'DEAMON' is 'if-needed'. This particular option has been used to generate the perimeter of a space whenever a space is created. Note that the difference between the 'DEFAULT' and 'DEAMON' functions is one of complexity. The 'DEFAULT' can only identify a constant to be used for a value. The 'DEAMON' can cause arbitrarily complex calculation to determine a value.

The instances of classes are used to represent architectural objects. For example, the first drawing action of the designer might be to sketch a rectangle which is then labeled 'office'. As the lines that make the four walls for this space are drawn, the system generates facts that describe what is represented by the drawing, in terms of the frames that have been defined for architectural objects. When the connection of the four walls, '21', '42', '35', and '27' is completed, the system will determine that a 'space' has been created and the instance for a 'space' frame would have been generated as seen in Table 2.

(	FRAME	space	243	)	
(	VALUE	space	name	243	office
(	VALUE	space	floor-height	243	1
(	VALUE	space	ceiling-height	243	8
(	VALUE	space	area	243	160
(	RELATION	space	wall	243	21
(	RELATION	space	wall	243	42
(	RELATION	space	wall	243	35
(	RELATION	space	wall	243	27
(	RELATION	space	symbol	243	22
(	VALUE	space	perimeter	243	52

**Table 2.** Modification of a 'space' Instance

## **The Semantic Network**

The frames used to represent architectural objects are referred to as design object frames and the collection of all such frames used by the Expert Design Advisor is called the Semantic Network. Thus, it is the values of the design object frames that represent the current state of the design solution within the context of the project.

As the designer draws within the CAD system, additional frames are automatically and transparently generated by the system to describe the architectural objects being represented and the relations between them. The frame information is generated in a blackboard environment to which the domain experts and the coordination expert are connected. As a result, the rules in any domain expert can fire as soon as their requisite facts are posted to the blackboard. This in turn might result in the posting of advice, in the form of new frame information, that could instantiate rules within the coordination expert.

## **Geometric Design Objects**

The architectural objects currently represented by the system include the following seven geometric design objects:

FLOOR, SPACE, WALL, DOOR, WINDOW, SEGMENT, and SYMBOL.

The FLOOR object refers to the level of a building, such as the first or second "floor". The SPACE object refers to an area, such as a room or duct. WALL, DOOR, and WINDOW have the obvious reference values.

The SEGMENT object refers to any part of a WALL object that is demarcated either by the intersection of another wall or has been drawn by the designer as a distinct wall component. The SYMBOL object represents directly by name any closed shape or icon within a SPACE object (eg., column, chair, table).

Since the focus of the ICADS system is on these design objects, it is natural to evaluate the use of COOL, the CLIPS Object Oriented Language (NASA 1991), to provide an "object" representation as a replacement for the frame representation that is in current use.

## **The Blackboard Coordination Expert**

The principal purpose of the Blackboard Coordination Expert is to assert frame slots, representing the current state of the evaluation process performed by the domain experts, onto the Semantic Network resident in the Blackboard. To accomplish this, the Coordination Expert receives from the Message Router all of the frames which contain results generated by the domain experts. The values that identify the current state of the design fall into one of three basic categories: values which result from solutions proposed by a single domain expert; values which result from solutions proposed by several domain experts for a common current value; and, values which must be inferred from solutions proposed by several domain experts.

In the case of the first category, which represents solution values unique to a single domain expert, the Coordination Expert does not change the values proposed by the domain expert. The proposed solution values are simply asserted as current values into the appropriate frame slots. In the second category two or more domain experts propose differing values for the same solution parameter. In such direct conflict situations it is the responsibility of the Coordination Expert to either determine which of the values is most correct or to derive a compromise value.

The Coordination Expert incorporates resolution rule sets which determine the best current values from those proposed. There is a resolution rule set for each possible direct conflict. In the development of each rule an attempt has been made to achieve a desirable balance between the various design issues. At this level the Coordination Expert can be considered an expert with knowledge of how this balanced integration can be achieved.

For example, if the structural, thermal, and sound domain experts agree in their suggestions for the material to be used for a wall, there is a rule in the Coordination Expert that will assign the common value as the wall's material in the solution value, which is referred to as the 'current value', or 'Blackboard value'.

With the frame representation that rule is written as shown in Table 3.

(defrule m2-floor-external-wall-material-weight-rule-1				
(FRAME	floor	?floor)		
(RELATION	floor	appearance/image	?floor	?floor-rel-struct )
(FRAME	appearance/image	source-idt	?floor-rel-struct	)
(VALUE	appearance/image	source-idt	?floor-rel-struct	struct )
(VALUE	appearance/image	external-wall-material-weight		
	?floor-rel-struct	?material)		
(RELATION	floor	appearance/image	?floor	?floor-rel-therm )
(FRAME	appearance/image	source-idt	?floor-rel-therm	)
(VALUE	appearance/image	source-idt	?floor-rel-struct	thermal )
(VALUE	appearance/image	external-wall-material-weight		
	?floor-rel-thermal	?material)		
(RELATION	floor	appearance/image	?floor	?floor-rel-sound )
(FRAME	appearance/image	source-idt	?floor-rel-sound	)
(VALUE	appearance/image	source-idt	?floor-rel-sound	sound )
(VALUE	appearance/image	external-wall-material-weight		
	?floor-rel-sound	?material)		
(RELATION	floor	appearance/image	?floor	?floor-rel-BB )
(FRAME	appearance/image	source-idt	?floor-rel-BB	)
(VALUE	appearance/image	source-idt	?floor-rel-BB	)
(VALUE	appearance/image	external-wall-material-weight		
	?floor-rel-BB	~?material)		
=>				
( bb_assert				
	(MODIFY VALUE	appearance/image	external-wall-material-weight	
		?floor-rel-BB	?material) )	
(bb_end_message)				
)				

**Table 3. A Coordination Rule With Frames**

In particular, note that only the single "VALUE" fact for the material-weight slot, or fact, is changed. The rest of the facts that are used to make the frame representation for the appearance/image frame remain the same. In addition, there is no notion of "access" to the frame in order to make this change.

The alert reader might wonder about the following line, that appears as the second pattern from the bottom of the left hand side:

(VALUE appearance/image source-idt ?floor-rel-BB )

It might seem that the 'source-idt' value is missing. Well, it is! As a matter of history, a 'Blackboard value', or 'current value' does not have a source identified. It is simply understood to be the Coordination Expert.

## A COOL Approach

The purpose of the frame representation scheme is to allow the ICADS system to reflect an understanding of architectural objects that are known to the user. It seems natural that an object oriented language such as COOL should be able to provide the same type of representation facility in an equal or better manner. However, the notion of "object" is notorious for harboring a wealth of misunderstanding and misuse.

Before attempting a large scale replacement of the thousands of lines of code that use the frame representation in an ICADS system, a careful evaluation must be made. In addition to the use of frames in domain expert systems and the coordination expert the frame scheme is used for utility purposes, such as the recognition and distribution of information in the Message Router module of the Blackboard (Taylor and Myers 1990). The Message Router uses pattern matching to identify the frame information that is distributed to the units that connect to the Blackboard. When a unit, such as a domain expert, connects to the Blackboard it provides a list of the frame information it wishes to receive. Since most units that connect to the Blackboard need only a small fraction of the facts received by the Blackboard, this selective distribution significantly reduces the amount of information communicated over the network.

Since COOL does not yet support pattern matching over objects, the impact of replacing frames in their CLIPS fact form with COOL objects must also be examined with respect to the communication uses of frame information, as well. First, however, an examination of the most common uses of the frame information is made.

## Sample COOL Objects

For example, in Table 3 there are two type of frames, the floor frame and the appearance/image frame. Each instance of such frames can be represented in COOL by the instance of an object that is used to represent the same architectural objects as the frames were intended to encode. A class can be defined for each of these architectural object types and a COOL object can be made correspondingly for each instance of an example frame representation.

For example, the object for the appearance/image frame might be defined, partially as:

```
(defclass (APPEARANCE/IMAGE (is-a USER)
  (slot (FLOOR-ID))
  (slot (SOURCE-IDT))
  (slot (EXTERNAL-WALL-MATERIAL-WEIGHT)
)
```

Then if ?floor held the name of a particular FLOOR object, the following could create an APPEARANCE/IMAGE object to record the suggestion of 'heavy' for the external-wall-material-weight that is being made by the structural domain expert:

```
(assert (appearance/image
  =(make-instance (gensym) of APPEARANCE/IMAGE
    (FLOOR-ID ?floor)
    (SOURCE-IDT struct)
    (EXTERNAL-WALL-MATERIAL-WEIGHT heavy)
  )
)
```

This assertion would generate a fact of the form, (appearance/image genx), where 'x'

is some integer and names the object made by the 'make-instance'. This fact can be matched by patterns in particular of the form, (appearance/image, ?v), to provide a convenient way of finding the instance names of the appearance/image objects in the left hand side of CLIPS rules.

Furthermore, the variable in the above pattern can be constrained by the attributes of the objects so that only certain objects are remembered by the variable. For instance, the form might be: (appearance/image, ?v&:(eq (send ?v get-SOURCE-IDT) struct)) In this case, only the objects that are recommended by the structural expert, that is, which have 'struct' as the value in the SOURCE-IDT slot would be matched.

Unlike the frame representation, the objects that hold the 'current values', 'Blackboard values', will have their SOURCE-IDT slots set to 'Coord'. Thus, with the appropriate definitions of the classes needed, the rule shown in its frame representation form in Table 3 could be used with COOL objects for the same purpose as below:

```
(defrule m2-floor-external-wall-material-weight-rule-1
  (appearance/image ?BB&:(eq (send ?BB get-SOURCE-IDT) Coord) )
  (appearance/image ?floor1&:(and (eq (send ?BB get-FLOOR-ID)
                                     (send ?floor1 get-FLOOR-ID))
                                (eq (send ?floor1 get-SOURCE-IDT) struct) ))

  (appearance/image ?floor2&:(and (eq (send ?BB get-FLOOR-ID)
                                     (send ?floor2 get-FLOOR-ID))
                                (eq (send ?floor2 get-SOURCE-IDT) thermal) ))

  (appearance/image ?floor3&:(and (eq (send ?BB get-FLOOR-ID)
                                     (send ?floor3 get-FLOOR-ID))
                                (eq (send ?floor3 get-SOURCE-IDT) sound) ))

  (test (eq (send ?floor1 get-EXTERNAL-WALL-MATERIAL-WEIGHT)
            (send ?floor2 get-EXTERNAL-WALL-MATERIAL-WEIGHT)
            (send ?floor3 get-EXTERNAL-WALL-MATERIAL-WEIGHT) ) )

  (test (neq (send ?floor1 get-EXTERNAL-WALL-MATERIAL-WEIGHT)
            (send ?BB get-EXTERNAL-WALL-MATERIAL-WEIGHT) ) )
=>
  (bind ?wt (send ?floor1 get-EXTERNAL-WALL-MATERIAL-WEIGHT) )
  (bb-assert (MODIFY ?BB EXTERNAL-WALL-MATERIAL-WEIGHT ?wt) )
  (bb-end-message)
)
```

**Table 4.** The Rule from Table 3, With Objects

This form of the rule sends messages to the appearance/image objects that make suggestions for the same object instance of the floor class, by checking the FLOOR-ID slot of the objects to make sure they are the same. It further sends messages to determine that the three objects whose instance names are held in ?floor1, ?floor2, and ?floor3 are for objects that hold the suggestions from the structural, thermal, and sound domain experts, respectively. Then it checks to make certain that the suggestions for EXTERNAL-WALL-MATERIAL-WEIGHT from these three objects are the same value. Finally, it determines if the current Blackboard value for this architectural attribute, as held in the ?BB object, is different from the common value in the three domain objects.

For the action part of the rule, consideration must be given to the manner in which the distributed Blackboard works. The Blackboard objects are replicated in the appropriate units

that attach to it. Therefore the action of asserting a result from the coordination unit is effected by a message that is sent from the Blackboard to all units that use the information being posted. Generally the message action is to ADD, MODIFY, or DELETE a fact.

### **Distributed Blackboard Action**

In Table 3 the MODIFY message will be interpreted by each unit that receives it as a command to change the fact that holds the external-wall-material-weight slot value in the instance of the appearance/image frame identified by the value of the ?floor-rel-BB value to become the value of the ?material variable. The unit receiving the message simply retracts the current fact of this form and asserts the new one. Note that it does not change any of the rest of the facts that identify the frame.

In the COOL version of the rule shown in Table 4, an equivalent distributed effect can be achieved by having the objects that hold the Blackboard information, the information asserted by the Coordination Expert, duplicated in the other units by objects that have the same instance name. Thus in the rule in Table 4, the same instance name is being sent to all of the units. The MODIFY function in each receiving unit will then perform the following:

(send b-string put-EXTERNAL-WALL-MATERIAL-WEIGHT m-string )

where b-string is the value of ?BB and m-string is the value of ?wt  
in the message that is transmitted by the bb-assert function

### **Refreshing Rules**

One other consideration must be given to the object scheme. When a slot in an object changes, the rules that might use that slot are unaware of the change. There is no mechanism in CLIPS/COOL that automatically refreshes rules that reference slots in objects, when the slots change their values. Therefore, the action of changing a slot value, as shown in the above paragraph, is accompanied by a retraction and assertion of the associated CLIPS fact for the instance of the object, to make certain that all rules that reference the object will be readied to fire again.

## **CONCLUSION**

The current working model of the ICADS system has the ability to advise a designer in a real-time design activity. It also exhibits a considerable amount of knowledge about the non-geometric attributes of the objects drawn and the real environment for the design project.

The most difficult part of extending the system is the classic problem of knowledge acquisition, for the domain expert systems and even moreso, for the coordination expert. One of the current problems is the difficulty of maintaining the exact form for referencing the same architectural object in references developed by different people at different times in different modules of a highly distributed system. A number of investigations to helping reduce this problem are in process in the ICADS laboratory. However, it seems that a great deal of help could be provided by the use of a representation for the architectural objects that better encapsulates the architectural object information into the expert system environment than the current frame representation protocol. COOL objects provide this improvement.

The use of COOL objects can also provide some speedup in the execution of certain computation. For example, in the current ICADS system the perimeter of a space is determined from the wall length information that is held in the frames for the walls that make up the space. When a wall is added in the drawing, pattern matching is used to identify the space to which the wall belongs. Then the old area fact is retracted and the updated area value is asserted in a new fact. By keeping the instance name of the space in each wall object, a method can be used to update the area slot in the space object whenever a change in

the wall object occurs. This particular action reduces both the pattern matching and retraction operations, which are fairly expensive activities.

In addition, it is easier and more efficient to implement in COOL objects the 'DEFAULT' and 'DEAMON' features from the current frame-based representation.

On the other hand, it can be seen by the example of Table 4 that the rules that reference the objects do not have an efficient way of getting the information from objects so that it can be used in the pattern matching process of their left hand sides. Performance evaluations on equivalent uses of larger scale will have to be done to identify whether a movement of the entire ICADS system to COOL is justified by the esthetic and maintenance advantages of working with COOL objects over working with the current frame-based representation.

## REFERENCES

- Assal, H. and Myers, L. (1990). An implementation of a frame-based representation in CLIPS, *First CLIPS Conference Proceedings*, Houston, pp.570-580.
- Myers, L. and Pohl, J. (1989). ICADS DEMO1: a prototype working model, *Fourth Eurographics Workshop on Intelligent CAD Systems*, Paris, France, pp.33-71.
- NASA (1989). *CLIPS Architecture Manual (version 4.3)*, Artificial Intelligence Center, Lyndon B. Johnson Space Center, Houston.
- NASA (1991). *CLIPS Architecture Manual (version 5.0)*, Artificial Intelligence Center, Lyndon B. Johnson Space Center, Houston.
- Pohl, J., L. Myers, A. Chapman and J. Cotton (1989). *ICADS: Working Model Version 1*, Technical Report: CADRU-03-89, CAD Research Unit, Design Institute, School of Architecture and Environmental Design, California Polytechnic State University, San Luis Obispo, California.
- Taylor, J. and Myers, L. (1990). Executing CLIPS expert systems in a distributed environment, *First CLIPS Conference Proceedings*, Houston, pp.686-697.